

PC printer port programs PROMs

Yongping Xia, EBT Inc, Torrance, CA



Fig 1 shows a 2817 programmer you can build. A PC's printer port controls the programmer. The printer port has eight data-output lines (pins 2 to 9), four control lines (pins 1, 14, 16, and 17), and five input lines (pins 10, 11, 12, 13, and 15). The data lines send out the 2817's address and program data. The control lines control the programming sequence. After programming is completed, the PC reads

the programmed data back in through its input port for verification.

The Turbo C program in Listing 1 sends the 2817's program, contained in a binary file named p2817.dat, to the programmer in this sequence: 8 bits of data, followed by the low 8 bits of the address, and finally by the high 3 bits of the address (which the programmer latches into IC₁, IC₂, and IC₃). The printer port's

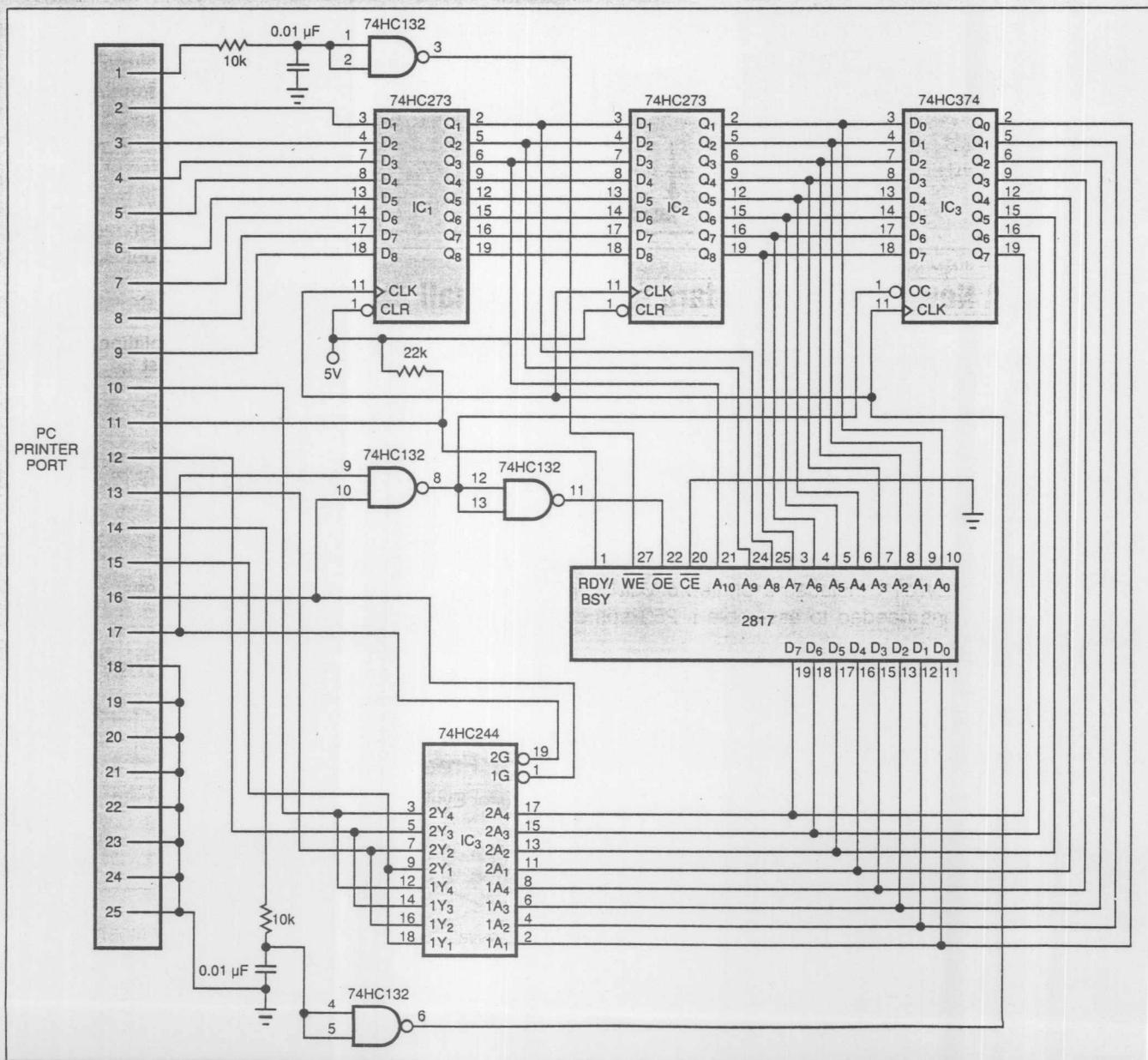


Fig 1—A PC's printer port and a simple Turbo C program control this 2817 programmer.

HIGH POWER FACTOR

Custom Switching Power Supplies
for ●Engineering work stations

- Computer main frames
- Computer peripheral equipment
- Business equipment
- Telecommunications
- etc.



- High power factor 0.99
- Design, manufacturing in Japan
- Repair center in U.S.A.

FDK also specializes in DC-DC converters, hybrid ICs, memory cards, ferrite cores, lithium batteries, stepper motors, optical isolators, etc.

FDK Your Best Strategic Partner
FDK AMERICA, INC.

A Division of Fuji Electrochemical Co., Ltd.
San Jose Office: 3099 North First Street, San Jose, CA 95134, U.S.A.
TEL: 408-432-8331 FAX: 408-435-7478

Dallas TEL: 214-650-7742 FAX: 214-650-7792
Boston TEL: 617-487-3198 FAX: 617-487-3199

CIRCLE NO. 70

Die cut components for pagers, cellular phones & computers



Scratch Resistant Lenses
EMI/RFI Shielding
Water Repellent Speaker Felts
Gaskets

Parts from prototype through production • Rolls or individual pieces
With or without adhesive • JIT delivery worldwide • Quality parts on time

the hirol company

1170 West McNab Road • Fort Lauderdale, Florida 33309 U.S.A.
001 (305) 974-4900 • Fax 001 (305) 975-4734

Read Cosine Limited (London)
0202 632274 or 632275
0202 623338 Day Fax
0202 600674 Night Fax

Dynamar Computer Products
Pte Ltd. (Singapore)
(65) 2813388
(65) 2813308 Fax

CIRCLE NO. 71

EDN-DESIGN IDEAS

pin 14 clocks these 19 bits in. Then the PC polls the 2817's RDY/BUSY line via printer-port pin 11, waiting for the chance to send the next data-address combination. After programming all 2 kbytes of the PROM, the PC will read the 2817 using printer-port pins 16 and 17. A copy of the program is available on the EDN BBS. EDN BBS /DL_SIG #1146 **EDN**

To Vote For This Design, Circle No. 313

Listing 1—2817 programmer control program

```
#include <stdio.h>

#define CONTROL_PORT 0x37a /* printer control port address */
#define IN_PORT 0x379 /* printer input port address */
#define OUT_PORT 0x378 /* printer output port address */

main()
{
    int buffer, tp, data[2048], i, add_h, add_l, readin_1, readin_2, read_data;
    FILE *stream;
    outportb(CONTROL_PORT, 0x07);
    if ((stream = fopen("p2817.dat", "rb")) == NULL)
    {
        fprintf(stderr, "Cannot open input file.\n");
        return 1;
    }
    fseek(stream, 0, 0);
    for (i=0; i<2048; i++)
    {
        buffer=0;
        fread(&buffer, 1, 1, stream);
        data[i]=buffer;
    }
    fclose(stream);
    for (i=0; i<2048; i++)
    {
        add_h=i/256;
        add_l=i-add_h*256;
        outportb(OUT_PORT, data[i]); /* send out data */
        outportb(CONTROL_PORT, 0x05);
        delay(1);
        outportb(CONTROL_PORT, 0x07);
        delay(1);
        outportb(OUT_PORT, add_l); /* send out low address */
        outportb(CONTROL_PORT, 0x05);
        delay(1);
        outportb(CONTROL_PORT, 0x07);
        delay(1);
        outportb(OUT_PORT, add_h); /* send out high address */
        outportb(CONTROL_PORT, 0x05);
        delay(1);
        outportb(CONTROL_PORT, 0x07);
        delay(1);
        outportb(CONTROL_PORT, 0x06); /* write address into 2817 */
        delay(1);
        outportb(CONTROL_PORT, 0x07); /* write data into 2817 */
        do {
            delay(1);
        } while (inportb(IN_PORT) & 0x80 != 0x00); /* wait */
    }
    for (i=0; i<2048; i++) /* check programmed data */
    {
        add_h=i/256;
        add_l=i-add_h*256;
        outportb(OUT_PORT, add_l); /* send out low address */
        outportb(CONTROL_PORT, 0x05);
        delay(1);
        outportb(CONTROL_PORT, 0x07);
        delay(1);
        outportb(OUT_PORT, add_h); /* send out high address */
        outportb(CONTROL_PORT, 0x05);
        delay(1);
        outportb(CONTROL_PORT, 0x07);
        delay(1);
        outportb(CONTROL_PORT, 0x03);
        delay(1);
        readin_1=inportb(IN_PORT) & 0x78; /* read low 4-bit data */
        outportb(CONTROL_PORT, 0x0f);
        delay(1);
        readin_2=inportb(IN_PORT) & 0x78; /* read high 4-bit data */
        outportb(CONTROL_PORT, 0x07);
        delay(1);
        read_data=readin_1/8 + readin_2*2;
        if (read_data != data[i])
        {
            fprintf(stderr, "Program error.\n");
            return 1;
        }
    }
    return 0;
}
```